SPECIFICATION

Please replace paragraph **[0002]** with the amended paragraph below.

[0002] Current satellite communication systems rely on bulk bandwidth allocation. The current Single Carrier per Channel (SCPC) and Time Division Multiplexing (TDM) systems allocate a fixed bandwidth for the duration of a ~~users~~user's session. This requires that the user actively initiate and terminate a data transfer session. For this reason, the bandwidth utilization can vary drastically based on the type of activity being performed. For rapid transfer of data the bandwidth has to be over allocated and under utilized.

Please replace paragraph **[0003]** with the amended paragraph below.

[0003] Communication over satellites is characterized by large propagation delays[[.]] ([[For]]for a geostationary satellite, the ~~one-way delay between two users or the two~~one-way delay between a user and the satellite or the two-way delay between two users is a minimum of 250 milliseconds[[.]]). Yet, users must share a common resource: the uplink bandwidth. Just as communications between users is subject to large delays, so also is the communication between users and the bandwidth manager (BWM) responsible for allocating uplink bandwidth among users.

Please replace paragraph **[0005]** with the amended paragraph below.

[0005] The CIR approach is wasteful of bandwidth in several ways. It ignores the fluctuation in use of bandwidth caused by users downloading or uploading and then pausing between operations. It ignores the asymmetry in consumption of user uplink bandwidth between downloads and uploads[[.]] (~~When~~when the user is downloading a file, he needs to uplink only an occasional acknowledgement back to the sender, but when he is uploading a file, he will uplink large amounts of data[[.]]). It ignores the variations in bandwidth utilization that occur over the course of a single upload or download[[.]] (~~Establishment~~establishment and termination of a file transfer use very small amounts of bandwidth compared with the bandwidth used to

transfer the file[[.]]). However, the CIR approach provides very good quality of service (QoS), because the user always has as much bandwidth as he could have expected.

Please replace paragraph **[0006]** with the amended paragraph below.

[0006] The bandwidth-on-demand approach, on the other hand, is very efficient in its allocation of bandwidth. In this approach, users request bandwidth only when they have data to send, and they request only as much bandwidth as they need to send their backlogged data. Thus, almost all the allocated bandwidth is actually used. However, the bandwidth on-demand approach can drastically cut throughput, and the QoS as perceived by the user can be terrible. The system operates in fits and starts, as users have backlogged data, request a limited amoutn of short-term bandwidth, wait through the delay to get bandwidth, send data, and then repeat the process. While[[,]] there are methods that partially ameliorate the problems with the bandwidth-on-demand approach, its service quality still remains poor.

Please replace paragraph **[0016]** with the amended paragraph below.

[0016] Fig. 7 is a flow diagram illustrating a mode of operation during which a user terminal sets an inactivity timeout so that when a data transfer is completed, the timeout elapses and ~~release~~releases bandwidth.

Please replace paragraph **[0021]** with the amended paragraph below.

[0021] 1) Dedicated allocation for dedicated services: No remote allocation approach can provide the required QoS for VoIP or IP teleconferencing without a dedicated allocation. The VBA gives these applications a priority service. The VBA makes this allocation on the basis of the ~~RSVP~~resource reservation protocol (RSVP) request for this type of service.

Please replace paragraph **[0024]** with the amended paragraph below.

[0024] 4) Fair Share Allocation: When the VBA allocates bandwidth to active users, it will allocate all available bandwidth based on the ~~users~~users' needs and fair share of available

bandwidth. This allows user sessions to complete earlier in periods of low activity and allows the system to support more than the normal user maximum for short bursts during periods of ~~haigh~~high user activity.

Please replace paragraph **[0026]** with the amended paragraph below.

[0026] 6) Combined bandwidth request/initial packet: For the TCP connection open, there may be enough space in the bandwidth request slot to hold both the bandwidth request and the initial TCP/IP SYN packet. In this case, the VBA would have the user terminal include ~~this~~these both in the bandwidth request slot. This combination of messages shortens a user's session by one round trip.

Please replace paragraph **[0029]** with the amended paragraph below.

[0029] The preferred embodiments address the problem of efficient allocation of return bandwidth in a satellite-based communications architecture. It also addresses the problem of providing to a user an "always-connected" paradigm rather than force the user to initiate all transfers with a dial up like activity. The delays created by satellite communication preclude the normal always-connected bandwidth on demand techniques seen in local systems like a ~~DOCSIS~~data over cable service interface specification (DOCSIS) based cable network. The current art for satellite communication systems is user-initiated, connection-oriented, bulk-bandwidth allocation.

Please replace paragraph **[0032]** with the amended paragraph below.

[0032] VBP includes the ability to support commitment to RSVP services, which may be required by user agreements. [[(]]RSVP is a very strict demand for fixed bandwidth. Voice over IP (VoIP) is expected to use RSVP to ensure that Internet phone calls do not experience unacceptable delay or packet loss.[[)]] If the system were operating as a simple "best effort" provider, we could accept short intervals of overloading where the system performance of specific connections is degraded. RSVP cannot accept such a degradation. VBP "fences off"

bandwidth committed via RSVP, ensuring that all guarantees to those sessions are met. It treats these high QoS sessions the same way that CIR treats them. For the rest of the sessions, VBP applies the bandwidth-efficient, good QoS approach described below.

Please replace paragraph **[0046]** with the amended paragraph below.

[0046] The VBP takes advantage of this situation by having the user terminal include its current total bandwidth need in the initial bandwidth request. The BWM[[,]] will allocate extra bandwidth (at the user terminal's full share) for a short time to allow the user terminal to clear its buffer, and then the BWM will reduce the allocation back to the minimum rate. [[(]]Obviously, if the data in the user terminal's input buffer is a small amount, the system will start the user at the minimum without the need for this early burst of higher rate bandwidth.[[)]] This approach allows the system to provide better performance to the users without tying up large amounts of bandwidth. If the user terminal is at the beginning of a TCP connection, this allocation profile matches the IP activity profile. If the user terminal is in the middle of a bursty client/server connection, this profile resembles bandwidth on demand.

Please replace paragraph **[0048]** with the amended paragraph below.

[0048] Based on the "Full allocation when loaded" strategy above, the user terminal can be in one of three allocation states: full return bandwidth, minimum return bandwidth, and no return bandwidth. The last opportunity to recover unused bandwidth is the procedure used to transition from full to minimum to no bandwidth. Normally, the user terminal releases the full bandwidth when its buffers have been clear for a given period of time, T1, which is a full bandwidth shut down lag time. The value of T1 will be fairly small, in order to keep bandwidth waste low. The user terminal releases the minimum bandwidth after it has been idle for a longer period, T2. [[(]]The minimum-to-no bandwidth transition lag, T2, must necessarily <u>be</u> held off for a longer time to allow for the delays in DNS and TCP conversations.[[)]]

Please replace paragraph **[0049]** with the amended paragraph below.

[0049] The values of [[Tl]]T1 and T2 can be dependent on current system data loading. The user terminal constantly computes when [[it]]its buffers will clear based on the current amount of backlogged data and the current bandwidth allocation. The BWM also provides the values of the lag times T1 and T2 to the user terminal. The user terminal calculates the point in time to initiate the release of uplink bandwidth based on the projected time when its buffer will clear and on lag times T1 and T2. The user terminal sends the full-to-minimum bandwidth transition request in anticipation of emptying its buffer and then having no further data to transmit for another [[Tl]]T1 seconds. If more data arrives in the terminal's input buffers or its allocation rate changes, the terminal recalculates its projected transition time and sends a countermanding transition request. The BWM always regards only the last-received transition request as valid. The reason for this process is to limit the time that the terminal wastes a full bandwidth allocation. If the terminal waited until its buffer was empty and then waited some additional lag time [[T l']]T1' before sending a transition request, then the fullrate bandwidth allocation could not end sooner than [[T 1']]T1' + RTT[[.]] (RTT is the round-trip time between the terminal and the BWM[[.]]). The VBP algorithm can cut the duration during which full bandwidth is wasted from T1' + RTT down to an arbitrarily small time. Further, this time can depend on the network conditions, with the time being longer when the network is lightly loaded and shorter when the network is heavily loaded. The BWM also can apply bandwidth release parameters to determine when bandwidth for an individual is released. Examples of bandwidth release parameters are:

Please replace paragraph **[0051]** with the amended paragraph below.

[0051] Referring to Fig. 1, a preferred form of the invention includes a processing or transponding satellite 100 usually in a geostationary orbit. Satellite 100 receives data from multiple user terminals [[200]] in a frequency division multiplexing (FDM), time division multiplexing (TDM) formatted stream. An exemplary user terminal 201 is one of multiple user terminals[[200]]. The user data is transmitted to an IP gateway 400 via satellite 100 by a

~~Return~~return link 210 ~~includes~~including an uplink 202 and a downlink 203. An uplink unit 204 in terminal 201 transmits the data to satellite 100 on a beam B1 forming part of uplink 202. Fig. 2 shows details of the uplink of return link 210. User terminal (UT) 201 contains a bandwidth requestor 220 and an IP based interface to a user ~~ADPE~~Automatic Data Processing Equipment (ADPE) 300, such as a personal computer (PC).

Please replace paragraph **[0052]** with the amended paragraph below.

[0052] Return link 210 is forwarded by satellite 100 to an IP gateway 400. The gateway extracts bandwidth requests for its local bandwidth manager 420 and sends the user's IP stream on to connected IP services 500. IP data is transmitted to multiple user terminals [[200]] via satellite 100 by a forward link 410 that includes an uplink 402 and a downlink 403. An uplink unit 404 in IP gateway 400 transmits the IP data to satellite 100 on a beam B2 forming part of uplink 402. Since there is a single source of data, the IP gateway's forward link 410 can be a single broadcast stream which may be formatted in any suitable manner for carrying IP data (i.e., MPEG using the DUB-S standard, ATM, or a special purpose packet format). The bandwidth allocations provided by the bandwidth manager 420 at the IP gateway 400 are multiplexed into the forward link 410 with the IP data.

Please replace paragraph **[0053]** with the amended paragraph below.

[0053] Fig. 2 shows the preferred form of the uplink manager with the variable bandwidth protocol (VBP). The uplink 202 of return link 210 can be any combination of a frequency division multiplexing (FDM) and time division multiplexing (TDM) format with FDM/TDM data cells, such as cell 211 (Fig. 2), that can individually be allocated to multiple user terminals[[200]]. The frequency divisions can be as few or as many as are possible within the allocated spectrum and the capabilities of the multiple user terminals[[200]]. The time division should allow an allocation of a fraction of the frequency to allocation of all the cells, such as cell 211, in an FDM division for a period of time. The nominal approach is to create repeating master frames of over four TDM slots per frame and allocate uplink by frequency,

frame position and starting and end frame IDs. User terminals with no current uplink bandwidth allocation use shared cells and a slotted aloha access technique 212 (Fig. 3) to request bandwidth. User terminals operating at minimum uplink bandwidth allocation [[212]]213 (Fig. 3) are allocated one cell per master frame. User terminals operating at full bandwidth 214 (Fig. 4) are allocated multiple cells per frame.

Please replace paragraph **[0054]** with the amended paragraph below.

[0054] Still referring to Fig. 2, each individual FDM/TDM data cell can be allocated separately and contains either a portion of the IP data transfer by the satellite terminal or a ~~Bandwidth~~bandwidth allocation request 211. Cells can be allotted to initial bandwidth requests for multiple user terminals using a slotted aloha access technique 212. The bandwidth manager (BWM) can ~~allocated~~allocate to a user terminal a single cell per frame for minimum bandwidth allocation 213. ~~the~~The BWM can allocate to a user terminal multiple cells per frame for fair share bandwidth allocation 214.

Please replace paragraph **[0070]** with the amended paragraph below.

[0070] In step 22, PC 300 begins a data transfer. Awaiting the TCP ACKs[[ ]], the PC 300 continues to send the packets.

Please replace paragraph **[0071]** with the amended paragraph below.

[0071] In step 23, terminal (UT) 201 begins the transfer of the packets using the minimum return bandwidth 213. It[[ ]] constantly computes when it will finish with the current data based on the current transmission rate.